Joint Modeling of Trigger Identification and Event Type Determination in Chinese Event Extraction

LI Pei Feng, ZHU Qiao Ming, DIAO Hong Jun and ZHOU Guo Dong[*]
School of Computer Science & Technology, Soochow University, Suzhou, China, 215006
{pfli, qmzhu, hjdiao, gdzhou}@suda.edu.cn

ABSTRACT

Currently, Chinese event extraction systems suffer much from the low quality of annotated event corpora and the high ratio of pseudo trigger mentions to true ones. To resolve these two issues, this paper proposes a joint model of trigger identification and event type determination. Besides, several trigger filtering schemas are introduced to filter out those pseudo trigger mentions as many as possible. Evaluation on the ACE 2005 Chinese corpus justifies the effectiveness of our approach over a strong baseline.

一个应用于中文事件抽取的事件触发词识别和类型判别联合模型

当前，有2个问题困扰着中文事件抽取系统：低质量的事件标记语料库和假事件触发词相对于真事件触发词的高比例。为了解决以上2个问题，本文提出了一个结合事件触发词识别和事件类型判别的联合模型。另外，几个触发词过滤模式同样被引入本系统用于过滤掉尽可能多的假触发词实例。在ACE2005中文语料上的测试结果表明，本文的方法和基准系统相比具有更高的性能。

KEYWORDS: Joint modeling, Event type determination, Trigger identification, Trigger filtering.
KEYWORDS IN L2: 联合模型, 事件类型判别, 触发词识别, 触发词过滤

[*]Corresponding author

1 Introduction

Information extraction (IE) is a task of extracting structured information (e.g. entities, relations and events) from the text. As a critical part of IE, event extraction is to identify trigger mentions of a predefined event type, and their participants and attributes. It can be typically divided into four components: trigger identification, event type determination, argument identification and argument role determination. Due to the central role of the contained events in a text, it is critical to mine their semantics in order to understand a text. Unfortunately, event extraction has been proven its performance is still very low.

In the literature, most studies focus on English event extraction and have achieved certain success (e.g., Grishman et al., 2005; Ahn, 2006; Patwardhan and Riloff, 2009; Hong et al., 2011; Lu and Roth, 2012; Llorens et al., 2012). However, there are few successful stories regarding Chinese event extraction due to the special characteristics in Chinese trigger identification. Besides unknown triggers[1] and word segmentation errors (Li et al., 2012), the low quality of annotated corpora and the high ratio of pseudo trigger mentions to true ones are also blamed for the low performance of Chinese event extraction.

To examine the low quality of annotated corpora in Chinese event extraction, we take the ACE (Automatic Content Extraction) 2005 Chinese corpus (with 8 types and 33 subtypes of events), one of the most popular corpora in event extraction, as an example. In particular, we randomly select 33 documents from the training set and ask two human annotators to manually tag event mentions and their types following the definition of the ACE 2005 corpus. Here, human annotator 1 is a first year postgraduate student with no background in Chinese event extraction while human annotator 2 is a third year postgraduate student working on Chinese event extraction. Table 1 justifies the difficulty of Chinese event extraction, particularly for trigger identification and event type determination, even for a well-educated human being. As shown in Table 1, the IAA (Inter-Annotator Agreement) on both trigger identification and event type determination is well below 50%. Even so, it is not surprising since the IAA on trigger identification on the ACE 2005 English corpus is only about 40% (Ji and Grishman, 2008).

Performance / Human	Trigger identification			Event type determination		
	P%	R%	F1	P%	R%	F1
annotator1 (blind)	63.3	62.9	63.1	61.7	59.5	60.6
annotator2 (familiar)	72.6	74.3	73.4	69.1	70.2	69.6
Inter-Annotator Agreement	45.8	42.9	44.3	45.3	42.5	43.8

TABLE 1 – Low quality of human annotation in the ACE 2005 Chinese corpus

Detailed analysis shows that one major reason for the low quality of human annotation is due to the difficulty of following the specified annotation guidelines, as mentioned in Ji and Grishman (2008). To better justify this issue, we randomly select 20 triggers and extract all the sentences which contain those triggers from the training set. Our exploration shows that although almost all the annotated trigger mentions are true ones, ensuring the reliability of the annotated trigger mentions, many true trigger mentions, e.g., those with exactly the same constituent or dependency structure as annotated ones, are not annotated, accounting for about 10% of trigger mentions. Table 2 shows the statistics.

[1] A trigger word/phrase occurring in the training data is called a known trigger and otherwise, an unknown trigger.

#Triggers	#Sentences	#Annotated trigger mentions	#un-annotated trigger mentions
20	452	198	23

TABLE 2 – Statistics of annotated vs. un-annotated trigger mentions in the ACE 2005 Chinese corpus

Take following two sentences as examples:

*(E1) 3 名抗议者在**冲突**中受伤。(Annotated trigger mention)*

*(Three protestors were injured in the **conflict**.)*

*(E2) 双方各有数人在**冲突**中受伤。(Un-annotated trigger mention)*

*(Several people from both sides were injured in the **conflict**.)*

Although the two examples are similar, “冲突” (conflict) in example (E1) is annotated as a trigger mention of the *Conflict* event while the one in example (E2) is not annotated. With the extreme example of “战争” (war), as the trigger of the *Conflict* event, among 11 trigger mentions concerned with “朝鲜战争” (Korean war) and “海湾战争” (gulf war), four of them are annotated as *Conflict* event while the others are ignored. Those un-annotated true trigger mentions would make the classifier difficult to distinguish true trigger mentions from pseudo ones.

For the high ratio of pseudo trigger mentions to true ones, Table 3 shows top 5 imbalanced triggers from the training set of the ACE 2005 Chinese corpus and justifies the difficulty for a classifier to identify a true trigger mention, especially for those of a particular event type, which appears only a few times in the training set.

Trigger[2]	#True trigger mentions	#Pseudo trigger mentions
投资 (invest)	1	67
建设 (set up)	1	66
取得 (obtain)	1	52
发 (provide)	1	36
给 (give)	2	64

TABLE 3 – Top 5 triggers with the highest ratios of pseudo trigger mentions to true ones in the ACE 2005 Chinese corpus

Recently, Li et al. (2012) justified that trigger identification was most critical for the performance of Chinese event extraction. In this paper, we also focus on trigger identification and its impact on overall Chinese event extraction.

In order to address the above-mentioned two critical issues in Chinese event extraction, this paper proposes a joint model of trigger identification and event type determination to improve the performance of trigger identification and overall Chinese event extraction. Besides, several trigger filtering schemas are introduced to filter out those pseudo trigger mentions as many as possible.

The rest of this paper is organized as follows. Section 2 overviews the related work. Section 3

[2] Most Chinese words have more than one sense. Here, we just give the one when it acts as a trigger.

describes the joint model of trigger identification and event type determination. Section 4 introduces those trigger filtering schemas. Section 5 evaluates our approach and shows its effectiveness over a strong baseline. Section 6 concludes the paper with future work.

2 Related work

To better understand the Chinese event extraction task as defined in ACE, where an event is defined as a specific occurrence involving participants, we list some ACE terminologies:

Event mention: a phrase or sentence within which an event is described, including a trigger and its arguments.

Trigger: the main word which most clearly expresses the occurrence of an event, so recognizing an event can be recast as identifying a corresponding trigger.

Trigger mention: a reference to a trigger word.

Trigger type/Event type: the type of an event.

Argument: the entity mentions involved in an event.

Argument role: the relation of an argument to an event where it participates.

In the literature, almost all the existing studies on event extraction are concerned with English. While earlier studies focus on sentence-level extraction (Grishman et al., 2005; Ahn, 2006; Hardy et al., 2006), later ones turn to employ high-level information, such as document (Maslennikov and Chua, 2007; Finkel et al., 2005; Patwardhan and Riloff, 2009), cross-document (Ji and Grishman, 2008), cross-event (Gupta and Ji, 2009; Liao and Grishman, 2010) and cross-entity (Hong et al., 2011) information.

2.1 Chinese event extraction

Compared with tremendous efforts in English event extraction, there are only a few studies on Chinese event extraction.

Some studies focused on feature selection. Tan et al. (2008) used a local feature selection method to ensure the performance of trigger classification and applied multiple levels of patterns to improve their coverage in argument classification. Fu et al. (2010) applied a feature weighting algorithm to re-weight various features extracted for trigger identification and event type determination. Chen and Ji (2009b) applied various kinds of lexical, syntactic and semantic features to address the special issues in Chinese. They also constructed a global errata table to record the inconsistency in the training set and used it to correct the inconsistency in the test set.

The other studies focused on automatic expansion of event triggers to improve the recall. Chen and Ji (2009a) proposed a bootstrapping framework, which exploited extra information captured by an English event extraction system. Ji (2009) first extracts some cross-lingual predicate clusters using bilingual parallel corpora and a cross-lingual information extraction system, and then employs the derived clusters to expand the triggers. Qin et al. (2010) described a method to expand the event triggers for Chinese event type determination based on a Chinese semantic dictionary “TongYiCi CiLin (expansion version)”. Li et al. (2012) proposed a novel inference mechanism to infer new trigger words by employing compositional semantics inside Chinese triggers. Their system achieved the state-of-the-art performance of 67.4 units in F1-measure on the ACE 2005 Chinese corpus, ignoring the post-processing – discourse consistency.

2.2 Joint modeling

While a pipeline model may suffer from the errors propagated from upstream tasks, a joint model can benefit from the close interaction between two or more tasks: it not only allows the uncertainty about one task to be carried forward to next ones but also allows useful information from one task to be carried backward to previous ones. Recently, joint modeling has been widely attempted in various NLP tasks, such as joint named entity recognition and syntactic parsing (Finkel and Manning, 2009), joint syntactic parsing and semantic role labeling (Li et al., 2010), joint anaphoricity and coreference determination (Denis and Baldridge, 2007; Iida and Poesio, 2011).

In the event extraction task, only a few studies are concerned with joint modeling, mostly in the bio-molecular domain. Riedel et al. (2009) used Markov Logic as a general purpose framework for jointly modeling the complete bio-molecular event structure for a given sentence. Poon and Vanderwende (2010) also adopted Markov Logic for bio-molecular event extraction in jointly predicting events and their arguments. Riedel and McCallum (2011) presented three joint models for bio-molecular event extraction. While the first model jointly predicts triggers and their arguments and the second model enforces additional constraints that ensure the consistency between events in hierarchical regulation structures, the third model integrates the first one and the second one in explicitly capturing the interaction of various arguments in the same event. Do et al. (2012) constructed a timeline of events mentioned in a given text which proposed a joint inference module that enforced global coherency constraints on the final outputs of the two pairwise classifiers, one between event mentions and time intervals, and one between event mentions themselves.

Our joint model is inspired by both Roth and Yih (2004) on joint named entity recognition and relation extraction and Denis and Baldridge (2007) on joint anaphoricity determination and coreference resolution. However, as far as we know, there are no successful models for jointly solving Chinese trigger identification and event type determination.

2.3 Trigger filtering

With the high ratio of pseudo trigger mentions to true ones, it is natural to filter out those unlikely trigger mentions in a preprocessing step. Basically, the general purpose for instance filtering is to reduce the class distribution imbalance by discarding harmful or superfluous instances.

In the literature, instance filtering has been widely employed in various NLP tasks. As for event extraction, there are also a few relevant studies. Patwardhan and Riloff (2009) first applied a self-trained relevant sentence classifier to identify relevant regions and split all candidate sentences into two sets: relevant and irrelevant sentences. Then, they used a pattern-based classifier to recognize events from those relevant sentences and a SVM-based classifier to recognize events from those irrelevant sentences. Landeghem et al. (2009) provided a negative-instances filter to check whether the length of the sub-sentence spanned by a candidate event does not exceed a certain value. Landeghem et al. (2010) further designed a false-positive filter using specific categories of relations to serve as negative indicators in Bio-NLP. Liao and Grishman (2010) applied a pseudo co-testing algorithm based on various criteria, such as informativeness, representativeness and diversity of the sentence, to filter out those pseudo samples to reduce annotation labour in event corpus annotation.

3 Joint modeling of trigger identification and event type determination

In this section, an ILP (Integer Logic Programming) -based inference framework is proposed to jointly model trigger identification and event type determination in reducing the influence of un-annotated true trigger mentions in the ACE 2005 Chinese corpus. Besides, a CRF (Conditional Random Field) model is applied as a supplement to the ME model to capture local sequential information around a trigger mention in trigger identification.

3.1 Joint inference of trigger identification and event type determination

As mentioned in Section 1, many true trigger mentions in the ACE 2005 Chinese corpus are not annotated. When training a classifier to identify trigger mentions, these un-annotated true trigger mentions in the training set will be extracted as negative samples. This will make the trigger identifier wrongly classify many true trigger mentions as pseudo ones, resulting in low recall in trigger identification. On the contrary, without the interference of these un-annotated true trigger mentions, the event type determiner has the higher probability of recognizing these annotated true trigger mentions as some kinds of events. This indicates the necessity and potential of jointly modeling for trigger identification and event type determination.

Besides, although the ME (Maximum-Entropy) model has been widely used in various subtasks of event extraction and achieved certain success in capturing the global information around a trigger mention, our experimentation shows that it suffers from low precision in trigger identification. To overcome this problem, a CRF model is introduced in trigger identification to capture the local sequential information. Our preliminary experimentation shows that the CRF model is much complementary to the ME model in trigger identification.

In our joint model, an ILP-based inference framework is introduced to integrate two trigger identifiers and one event type determiner. Figure 1 shows the ILP-based inference framework, which integrates a CRF-based trigger identifier (CRF_I) with an ME-based trigger identifier (ME_I) and an ME-based event type determiner (ME_D). The features used by ME_D and ME_I are as same as Li et al. (2012).

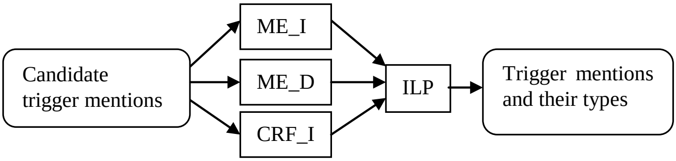


FIGURE 1 – Joint modeling of trigger identification and event type determination

ILP is a mathematical method for constraint-based inference to find the optimal values for a set of variables that minimize an objective loss function in satisfying a certain number of constraints. In the literature, ILP has been widely used in various NLP tasks (e.g., Roth and Yih, 2004; Barzilay and Lapata, 2006; Iida and Poesio, 2011; Do et al., 2012) in combining multiple classifiers, where the traditional pipeline architecture is not appropriate.

We assume $p_{ME\_I}(EVENT|Tr_{i,j})$ is the probability of ME_I identifying a trigger mention as a true one, where $Tr_{i,j}$ is the *jth* mention of the *ith* trigger word in a discourse, and $p_{ME\_d}(R_k\ |Tr_{i,j})$ *is* the probability of ME_D determining a trigger mention as an event of type R_k. Like Roth and Yih (2004), we define following assignment costs:

$$c^{I}_{<i,j>} = -\log(p_{ME\_I}(EVENT \mid Tr_{i,j}) \tag{1}$$

$$\bar{c}^{I}_{<i,j>} = -\log(1 - p_{ME\_I}(EVENT \mid Tr_{i,j}) \tag{2}$$

$$c^{D}_{<i,j>} = -\log(p_{ME\_D}(R_k \mid Tr_{i,j}) \tag{3}$$

$$\bar{c}^{D}_{<i,j,k>} = -\log(1 - p_{ME\_D}(R_k \mid Tr_{i,j}) \tag{4}$$

where $c^{I}_{<i,j>}$ is the cost of $Tr_{i,j}$ as an event trigger mention while $\bar{c}^{I}_{<i,j>}$ is the cost of $Tr_{i,j}$ not as an event trigger mention; $c^{D}_{<i,j>}$ is the cost of $Tr_{i,j}$ as an event trigger mention of type R_k while $\bar{c}^{D}_{<i,j,k>}$ is the cost of $Tr_{i,j}$ not as an event trigger mention of type R_k.

Besides, we use indicator variable $x_{<i,j>}$ that is set to 1 if $Tr_{i,j}$ is an event mention, and 0 otherwise. Similar to $x_{<i,j>}$, we use another indicator variable $y_{<i,j,k>}$ that is set to 1 if $Tr_{i,j}$ is an event mention of type R_k, and 0 otherwise. Finally, the objective function of the ILP-based inference framework can be represented as follows, where D is the set of trigger words in a discourse and M_i is the set of all mentions with the same *ith* trigger word,

$$\begin{aligned} \min \; & \sum_{i\in D}\sum_{j\in M_i}(c^{I}_{<i,j>}x_{<i,j>} + \bar{c}^{I}_{<i,j>}(1 - x_{<i,j>})) \\ & + \sum_{i\in D}\sum_{j\in M_i}\sum_{1\le k\le 33} c^{D}_{<i,j,k>}y_{<i,j,k>} + \bar{c}^{D}_{<i,j,k>}(1 - y_{<i,j,k>})) \end{aligned} \tag{5}$$

Subject to

$$x_{<i,j>} \in \{0,1\} \qquad \forall i \in D \wedge j \in M_i \tag{6}$$

$$y_{<i,j,k>} \in \{0,1\} \qquad \forall i \in D \wedge j \in M_i \wedge 1 \le k \le 33 \tag{7}$$

To enforce consistency, we add further constraints:

(C1) Event type constraint: if a trigger mention $Tr_{i,j}$ belongs to event type R_k, it must be a true trigger mention.

$$x_{<i,j>} \ge y_{<i,j,k>} \qquad \forall i \in D \wedge j \in M_i \wedge 1 \le k \le 33 \tag{8}$$

(C2) True trigger mention constraint: if a mention $Tr_{i,j}$ is a true trigger mention, it must belong to only one event type R_k.

$$x_{<i,j>} = \sum_{1\le k\le 33} y_{<i,j,k>} \qquad \forall i \in D \wedge j \in M_i \tag{9}$$

(C3) Discourse consistency: all trigger mentions which have the same trigger word must have the same event type in a discourse, or all of them aren't true trigger mentions.

$$x_{<i,j>} = x_{<i,l>} \qquad \forall i \in D \wedge j,l \in M_i \tag{10}$$

As a discourse-driven language, the syntax of Chinese is not as strict as English and very often we need to count on the discourse-level information to understand the meaning of a Chinese sentence. As for an event, a trigger may appear many times in a discourse and a trigger is considered discourse-consistent when all its appearances have the same event type. The statistics on the training sets of both the ACE 2005 Chinese and English corpora shows that within a discourse, there is a strong consistency in both Chinese and English between trigger mentions: if one instance of a word is a trigger, all the other instances in the same discourse will be a trigger of the same event type with a very high probability (> 90% in Chinese).

(C4) Different event types for trigger mentions in a clause: Different trigger mentions in a clause must have different event types.

$$y_{<i,j,k>} + y_{<r,t,k>} < 2 \quad \forall Tr_{i,j} \in cl_a \wedge Tr_{r,t} \in cl_a \wedge i,r \in D \wedge j \in M_i \wedge t \in M_r \wedge 1 \le k \le 33 \wedge Tr_{i,j} \ne Tr_{r,t} \qquad (11)$$

where cl_a is the set of words in clause a.

For example, trigger mentions “暴力” (violence, *Conflict* event) and “冲突” (conflict, *Conflict* event) may occur together to form a phrase “暴力冲突” and we should identify them as one *Conflict* event instead of two.

(C5) Cross-event constraint: Those events with high probability of co-occurring in a discourse must have same indicator values (event or non-event).

$$y_{<i,j,k>} = y_{<r,t,k^{'}>} \quad \forall i,r \in D \wedge j \in M_i \wedge t \in M_r \wedge <k,k^{'}> \in O \wedge 1 \le k \le 33 \wedge 1 \le k^{'} \le 33 \qquad (12)$$

where O is the set of event type pairs with high probability of co-occurring[3] in the training set.

As mentioned in Liao et al. (2010), there are strong correlations among event types in a document. We also find out that some events have a high probability of co-occurring in a discourse. For examples, if there is a *Die* event in a discourse, there is more than 70% probability that an *Attack* event also appears in the same discourse.

3.2 CRF-based trigger identification

CRF is a conditional sequence model which represents the probability of a hidden state sequence given some observations. It is a popular and efficient machine learning technique for supervised sequence labeling and has been applied to many NLP tasks.

We choose CRF due to its ability of capturing the local information around a trigger mention. For this purpose, we build a separate character-based trigger identifier and use the CRF model to label each character with a tag indicating whether it is out of a given trigger (O), the beginning of the trigger (B) or a part of the trigger except the beginning one (I). In this way, our CRF-based trigger identifier performs sequential labeling by assigning each character one of the three tags and a character assigned with tag B is concatenated with following characters with tag I to form a trigger. For example, example (E1) can be labelled as follows and***冲突*** is identified as a trigger.

(E3) 3/O 名/O 抗/O 议/O 者/O 在/O ***冲/B 突/I*** *中/O 受/O伤/O 。/O*

To achieve high precision as much as possible, we just use the character itself and characters around it as features. For each character c_i, assuming its 5-windows characters are $c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}\ c_{i+2}$, our CRF-based trigger identifier adopts following features: c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}, $c_{i-1}c_i$, c_ic_{i+1}, $c_{i-2}c_{i-1}c_i$, $c_{i-1}c_ic_{i+1}$, $c_ic_{i+1}c_{i+2}$.

Our preliminary experimentation shows that the CRF model achieves high precision and is much complementary to the ME model in trigger identification. In this paper, the CRF-based trigger identifier is included into the ILP-based inference framework by introducing one more constraint.

(C6) CRF trigger constraint: due to high precision of the CRF model, we include a simple inference rule in our joint model:

[3] The threshold of the probability of the event type pair is fine-tuned to 0.70 using the development set.

$$x_{<i,j>} = 1 \text{ if the CRF model identifies } Tr_{i,j} \text{ as an event} \quad (13)$$

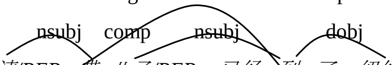
4 Trigger filtering

In this section, we firstly introduce two trigger filtering mechanisms: dependency-based inference mechanism and divide-and-conquer mechanism, to remove pseudo trigger mentions. Then we provide another trigger filtering mechanism by employing both local and global discrimination to filter out those un-annotated true trigger mentions.

4.1 Dependency-based inference mechanism

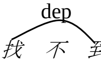
Some single-character trigger words are very ambiguous, e.g. with more than one senses and POSs (part of speeches), and it is hard to tell their true trigger mentions from their pseudo ones. For example, "到" (come) has 246 mentions (not including those words containing "到") corresponding to 6 senses and 3 POSs, and only 84 of them are true trigger mentions. In this paper, according to the total number of senses and POSs, we select 32 top ambiguous single-character words, such as "到", "往", "并", and employ a dependency-based inference mechanism to filter out their occurrences as pseudo trigger mentions.

In a sentence, there normally exists strong structural dependency between a trigger and its arguments. Take following two sentences as examples:



(E4) 朱婉清/PER 带 儿子/PER 已经 到 了 纽约/GPE。

(Zhu Wanqing has arrived New York with her son.)



(E5) 我/PER 找 不 到 工作。

(I can't find a job.)

(E4) is a positive example where "到" (arrive) is a trigger mention of the *Movement* event and there is a strong structural dependency between the trigger and its arguments, while example (E5) is a negative example where "到" is not a *Movement* event and there is no obvious dependency between the trigger and subject "我" (I).

In this paper, we adopt Markov Logic Network (MLN) to determine whether a single-character word is a trigger mention or not. For this purpose, we construct two inference formulas based on the dependency and POS information as follows, similar to Poon and Vanderwende (2010):

$$Token(i,+w) \wedge Pos(i,+o) => Event(i)$$

$$Token(j,+w) \wedge Dep(i,j,+r) => Event(i)$$

where

Token(i,w): whether token *i* has word *w*;

Pos(i, o): whether token *i* has POS *o*;

Dep(i, j, r): whether there is a dependency edge from *i* to *j* with relation *r* or relation path *r* (e.g., ccomp->nsubj, pp->pobj);

Event (i): whether token *i* is an event trigger mention.

Here, notation "+" signifies that the MLN contains an instance of the formula, with a separate weight, which is learnt from the training set. In particular, the open-source *Alchemy* package[4] is employed for learning and inference. Like Pooh and Vanderwende (2010), we use the Stochastic Gradient Descent (SGD) to learn weights and introduce MC-SAT, a slice sampling Markov chain Monte Carlo algorithm, to make the inference. To obtain a final assignment, we set the query atoms with probability no less than 0.3 (fine-tuned to maximize F1 on the development set) to true and the rest to false, in order to keep true trigger mentions.

4.2 Divide-and-conquer mechanism

Trigger mentions with high ratios of pseudo trigger mentions to true ones are treated differently from those with low ratios, using a threshold θ[5]. For the former, two patterns are applied to filter those high-unlikely trigger mentions as follows, while for the later, we adopt a ME classifier to filter out pseudo trigger mentions as many as possible.

(P1) <entity type of subject> <trigger[6]>

(P2) <trigger> <entity type of direct/indirect object>

where the subject and object must be the arguments of that event.

4.3 Local and global discrimination

Like the representativeness of an event, the *discrimination* considers the distributional similarity of a pseudo trigger mention against those true trigger mentions. If a pseudo trigger mention is similar to one of those true trigger mentions, it should be filtered out from the set of pseudo trigger mentions due to its low *discrimination*; otherwise, it should be kept in the set of pseudo trigger mentions due to its high *discrimination*. For an un-annotated true trigger mention which is extracted as a pseudo trigger mention, it tends to have the large distributional similarity with those true trigger mentions and should be filtered from the set of pseudo trigger mentions due to its low discrimination.

Normally, the distribution of events of a particular type is not balanced. For example, in the ACE 2005 Chinese corpus, *Movement* events occur most frequently in the training set with 701 times and occupy 22.0% of all event occurrences, while for the 10 least frequently-occurring event types (e.g. *Execute*, *Delare-Bankruptcy*, *Divorce*, etc.), each of them only occupies less than 1%.

To well address the above phenomenon, this paper introduces two types of discrimination, local discrimination *local_d* and global discrimination *global_d*, to filter out those un-annotated true trigger mentions and reduce their negative impact on trigger identification.

On the one hand, the local discrimination measures the similarity between a particular pseudo trigger mention and all of true trigger mentions with the same trigger word (shorted as STMs). In our case, each trigger mention is represented as a vector of features and the cosine similarity is applied to measure the similarity between a pseudo trigger mention and each STM.

If the pseudo trigger mention is similar to one STM, their similarity will be high. Instead of calculating the average similarity, we calculate the maximum similarity to identify whether the pseudo trigger mention should be filtered out:

[4] http://alchemy.cs.washington.edu/

[5] Threshold θ is fine-tuned to 3, using the development set.

[6] We just extract those trigger mentions with POS verb.

$$local\_d(\vec{p}) = \underset{i \le i \le n}{Max}(sim(\vec{p}, \vec{m_i})) \tag{14}$$

where n is the number of STMs for the pseudo trigger mention p and $sim(\vec{p}, \vec{m_i})$ is the cosine similarity between p and its STM m_i.

On the other hand, the global discrimination comes from the probability of a pseudo trigger mention belonging to the set of true trigger mentions in the training set. While the local discrimination measures the distance between a pseudo trigger mention and those STMs, the global discrimination calculates the distance between a pseudo trigger mention and all the true trigger mentions. In this paper, we use the probability from the event type determiner to calculate the global discrimination:

$$global\_d(\vec{p}) = \underset{1 \le i \le m}{Max}(p(R_i \mid \vec{p})) \tag{15}$$

where m is the number of event types, R_i is the type of a particular event and $p(R_i \mid \vec{p})$ is the probability from the event type determiner.

Given local and global discrimination, the final *discrimination* is calculated via linear interpolation.

$$\text{discrimin ation}(\vec{p}) = 1 - (\alpha * global\_d(\vec{p}) + (1-\alpha) * local\_d(\vec{p})) \tag{16}$$

where coefficient α (0≤α≤1) is fine-tuned to 0.75 on the development set and all trigger mentions whose discrimination values are lower than 0.1 are filtered out.

5 Experimentation and discussion

In this section, we evaluate our trigger filtering mechanisms and joint model in Chinese trigger identification and its application to overall Chinese event extraction.

5.1 Experimental setting

For fair comparison, we use the state-of-the-art Chinese event extraction system, as described in Li et al. (2012), as our baseline[7], which consists of four typical components, trigger identification, event type determination, argument identification and argument role determination, and works in a pipeline way. During testing, each word in the test set is first scanned for instances of known triggers from the training set and then scanned by employing the compositional semantics inside Chinese triggers to infer instances of unknown triggers. When an instance is found, the trigger identifier is applied to distinguish those true trigger mentions from pseudo ones. If true, the event type determiner is then applied to recognize its event type. For any entity mention in a sentence which is identified as an event, the argument identifier is employed to assign its possible arguments afterwards. Finally, the argument role determiner is introduced to assign a role to each argument.

Besides, we adopt the same experimental setting as Li et al. (2012) and all the evaluations are done on the ACE 2005 Chinese corpus (only the training data is available), which contains 633 Chinese documents annotated with 8 predefined event types and 33 predefined event subtypes. Similar to previous studies, we treat these subtypes simply as 33 separate event types and do not

[7] To simplify the experiments, the baseline only contains compositional semantics in Li et al. (2012).

consider the hierarchical structure among them. Particularly, we randomly select 567 documents as the training set and the remaining 66 documents as the test set. Besides, we reserve 33 documents in the training set as the development set and follow the setting of ACE diagnostic tasks and use the ground truth entities, times and values for our training and testing. As for evaluation, we also follow the standards as defined in Li et al (2012):

- A trigger is **correctly** identified if its position in the document matches a reference trigger;
- An event type is **correctly** determined if the trigger's event type and position in the document match a reference trigger;
- An argument is **correctly** identified if its involved event type and position in the document match any of the reference argument mentions;
- An argument role is **correctly** determined if its involved event type, position in the document, and role match any of the reference argument mentions.

Finally, all the sentences in the corpus are divided into words using a Chinese word segmentation tool (ICTCLAS[8]) with all entities annotated in the corpus kept. Besides, we use Berkeley Parser[9] and Stanford Parser[10] to create the constituent and dependency parse trees and employ the ME model[11] to train individual component classifiers.

5.2 Trigger filtering

Table 4 shows the impact of the three trigger filtering mechanisms in Chinese event extraction on the held-out test set. From Table 4, we can find out that our trigger filtering mechanisms enhance the F1-measures of trigger identification, event type determination, argument identification and argument role determination by 2.5, 2.7, 2.6 and 2.3 units, respectively. It justifies the effectiveness of our trigger filtering mechanisms in addressing the low quality of the ACE 2005 Chinese corpus.

Performance	Trigger identification			Event type determination			Argument identification			Argument role determination		
System	P%	R%	F1	P%	R%	F1	P%	R%	F1	P%	R%	F1
Baseline	73.5	62.1	67.4	70.2	59.1	64.2	58.0	48.9	53.0	54.7	44.5	49.1
+DepInference	75.4	61.9	68.0	71.9	59.1	64.9	59.8	48.9	53.8	56.1	44.5	49.6
+D&C	**76.7**	62.9	69.2	**73.1**	59.9	65.6	**60.7**	49.7	54.7	**57.2**	45.4	50.6
+L&G	75.0	**65.4**	**69.9**	71.6	**62.7**	**66.9**	59.5	**52.1**	**55.6**	56.1	**47.5**	**51.4**

TABLE 4 – Contribution of trigger filtering to Chinese event extraction (incremental)

Detailed analysis shows that

- The dependency-based inference (**DepInference**) filters out 8.5% of candidate trigger mentions and 98.8% of them are pseudo ones. As a result, Table 4 shows that this inference improves the precision by 1.9 units for trigger identification with only a slight loss of 0.2 units in the recall. Given the fact that only 20% trigger mentions are single-character words

[8] http://ictclas.org/
[9] http://code.google.com/p/berkeleyparser/
[10] http://nlp.stanford.edu/software/lex-parser.shtml
[11] http://mallet.cs.umass.edu/

and some of them can be distinguished by the trigger identifier, this justifies the effectiveness of the dependency-based inference in filtering out those pseudo single-character trigger mentions.

- While the divide-and-conquer mechanism **(D&C**) filters out 20.3% of candidate trigger mentions, it is surprising that less than 6% of filtered trigger mentions are true ones. As a result, the divide-and-conquer mechanism much improves the F1-measure, precision and recall by 1.3, 1.0 and 1.2 units respectively. Our exploration also shows that our two simple patterns can recover almost 40% of the filtered true trigger mentions.
- The local and global discrimination (**L&D**) improves the recall by 2.5 units with a loss of 1.7 units in precision. When coefficient α is fine-tuned to 0.75 and the threshold of $\text{discrimination}(\vec{p})$ is fine-tuned to 0.1, 5.8% of pseudo trigger mentions are filtered out, in which almost 80% of them are un-annotated true trigger mentions.

5.3 Joint modeling

Table 5 shows the contribution of both Trigger Filtering (**TF**) and Joint Modeling (**JM**) of trigger identification and event type determination to overall Chinese event extraction on the held-out test set. Table 5 indicates that our approach can improve the F1-measures of trigger identification, event type determination, argument identification and argument role determination (i.e. overall event extraction) by 5.7, 6.0, 5.1 and 4.8 units, respectively, largely due to the dramatic increase in recall of 9.8, 9.8, 8.3 and 7.1 units respectively.

Performance	Trigger identification			Event type determination			Argument identification			Argument role determination		
System	P%	R%	F1	P%	R%	F1	P%	R%	F1	P%	R%	F1
CRF	**83.7**	43.3	57.1									
Baseline	73.5	62.1	67.4	70.2	59.1	64.2	58.0	48.9	53.0	54.7	44.5	49.1
+JM(w/o CRF)	73.1	68.1	70.5	69.9	65.1	67.4	57.8	53.9	55.8	54.6	49.1	51.7
+JM(w/ CRF)	73.0	70.0	71.5	69.9	67.0	68.4	57.8	55.6	56.7	54.6	50.6	52.5
+TF+JM(w/ CRF)	74.4	**71.9**	**73.1**	**71.4**	**68.9**	**70.2**	**59.1**	**57.2**	**58.1**	**55.8**	**52.1**	**53.9**

Table 5 – Contribution of joint modeling to Chinese event extraction

Table 5 also shows that

- For trigger identification, the ILP-based joint model (w/o CRF) improves the F1-measure by 3.1 units due to a big gain of 6.0 units in recall and a small loss of 0.4 units in precision. This result indicates that event type determination can much help trigger identification to improve its performance. This justifies the effectiveness of our ILP-based joint model. As for the loss in precision, it's not surprising that more pseudo trigger mentions tend to be wrongly recognized as true ones since our goal of various constraints in the ILP-based joint model is to identify those true trigger mentions as many as possible.
- Further inclusion of the CRF model as a constraint in the joint model improves the F1-measure of trigger identification by 4.1 units due to a big gain of 7.9 units in recall and a small loss of 0.5 units in precision. Our experimentation also shows that the CRF model is much complementary to the ME model in trigger identification with a high precision of 83.7

units and a low recall of 43.3 units, and the new constraint helps bring back 1.9% of true trigger mentions.

- We also apply the postprocessing mechanism of discourse consistency in Li et al. (2012) to both the baseline and our approach, and their improvement of F1-measues in trigger identification are 3.1 units and 2.4 units respectively. The reason for the loss of our approach is that our three trigger filtering mechanisms reduce the probability of the consistency in a discourse for those filtered pseudo trigger mentions.
- Finally, our trigger filtering schema and joint modeling of trigger identification and event type determination together significantly improve the recall for all of four components in Chinese event extraction with a decent gain in precision.

5.4 Discussion

From Table 5, we can also find out that the performance gaps between trigger identification and event type determination are rather small in all settings (2.9~3.2 units in F1-measures). The fact is that, even if we just assign the type with the highest prior probability to all true trigger mentions, the accuracy can still reach more than 90%. This indicates the importance of trigger identification in overall Chinese event extraction.

Normally in a pipeline system, the improvement in event type determination is always lower than that in trigger identification due to the pipeline nature (i.e. propagated errors from the upstream processes). However, Table 5 shows that our improvement in F1-measure for event type determination is higher than that for trigger identification. This is due to joint modeling of these two components in well capturing the interaction between them.

Conclusion

In order to address the special characteristics of Chinese event extraction, this paper presents a joint model to better integrate trigger identification and event type determination. Besides, several trigger filtering mechanisms are proposed to reduce the influence of those un-annotated true trigger mentions in the corpus as many as possible. The experimental results show that our approach can significantly improve the performance of Chinese trigger identification and overall Chinese event extraction.

Besides those un-annotated true trigger mentions, which much encumber the performance of trigger identification and overall event extraction, we find that 9.7% of the pseudo trigger mentions in the ACE 2005 Chinese corpus are actually true ones. Therefore, a natural extension of this work is to explore some effective methods to recover those pseudo-annotated true trigger mentions. Moreover, encouraged by the success of the ILP-based joint model, we will further explore more on this joint model and more effective joint models to event extraction.

Acknowledgments

The authors would like to thank three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China under Grant Nos. 61070123, 61272260 and 61273320, the National 863 Project of China under Grant No. 2012AA011102.

REFERENCES

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating* *and Reasoning about Time and Events (ARTE 2006)*, pages 1-8.

Barzilay, R. and Lapata, M. (2006). Aggregation via set partitioning for natural language generation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2006)*, pages 359-366.

Chen, Z. and Ji, H. (2009a). Can one language bootstrap the other: a case study on event extraction. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing (SemiSupLearn 2009)*, pages 66-74.

Chen, Z. and Ji, H. (2009b). Language specific issue and feature exploration in Chinese event extraction. In *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2009), pages 209-212.

Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, pages 236–243.

Do, Q. X., Lu, W., and Roth, D. (2012). Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 677-687.

Finkel, J., Grenager, T. and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 363-370.

Finkel, J. and Manning, C. (2009). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 141-150.

Fu, J. F., Liu, Z. T., Zhong, Z. M. and Shan, J. F. (2010). Chinese event extraction based on feature weighting. *Information Technology Journal*, 9: 184-187.

Gupta, P. and Ji, H. (2009). Predicting unknown time arguments based on cross-event propagation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* (ACLShort 2009), pages 369-272.

Grishman, R., Westbrook, D. and Meyers, A. (2005). NYU's English ACE 2005 system description. In *Proceedings of ACE 2005 Evaluation Workshop (ACE workshop 2005)*.

Hardy, H., Kanchakouskaya, V. and Strzalkowski, T. (2006). Automatic event classification using surface text features. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence Workshop on Event Extraction and Synthesis (AAAI workshop 2006)*, pages 36-41.

Hong, Y., Zhang, J., Ma, B., Yao, J. M., Zhou, G. G. and Zhu, Q. M. (2011). Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 1127-1136.

Iida, R. and Poesio, M. (2011). A cross-lingual ILP solution to zero anaphora resolution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 804-813.

Ji, H. (2009). Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics (UMSLLS 2009)*, pages 27-35.

Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2008)*, pages 254-262.

Landeghem, S. V., Saeys Y., Baets, B. D. and Peer, Y. V. (2009). Analyzing text in search of bio-molecular events: a high-precision machine learning framework. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task (BioNLP 2009)*, pages 128-136.

Landeghem, S. V., Pyysalo, S., Ohta, T. and Peer, Y. V. (2010). Integration of static relations to enhance event extraction from text. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (BioNLP 2010)*, pages 144-152.

Li, J. H., Zhou, G. D. and Ng, H. T. (2010). Joint syntactic and semantic parsing of Chinese. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1108-1117.

Li, P. F., Zhou G. D., Zhu Q. M. and Hou L. B. (2012). Employing compositional semantics and discourse consistency in Chinese event extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 1006-1016.

Liao, S. S. and Grishman, R. (2010). Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 789-797.

Liao, S. S. and Grishman, R. (2011). Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 714–722.

Llorens, H., Saquete, E., Navarro-Colorado, B. (2012). Applying semantic knowledge to the automatic processing of temporal expressions and events. *Information Processing & Management*.

Lu, W. and Roth, D. (2012). Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (ACL 2012), pages 835–844.

Maslennikov, M. and Chua, T. (2007). A multi resolution framework for information extraction from free text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 592-599.

Patwardhan, S. and Riloff, E. (2007). Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-coNLL 2007)*, pages 717-727.

Patwardhan, S. and Riloff, E. (2009). A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 151-160.

Poon, H. and Vanderwende, L. (2010). Joint inference for knowledge extraction from biomedical literature. In *Proceedings of Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 813-821.

Qin, B., Zhao, Y. Y., Ding, X., Liu, T. and Zhai, G. F. (2011). Event type recognition based on trigger expansion. *Tsinghua Science and Technology*. 15(3): 251-258.

Riedel, S., Chun, H. W., Takagi, T. and Tsujii, J. (2009). A Markov logic approach to bio-molecular event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task (BioNLP 2009)*, pages 41-49.

Riedel, S. and McCallum, A. (2011). Fast and robust joint models for biomedical event extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1-12.

Roth, D. and Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In *Proceedings of the 8th Conference on Natural Language Learning (CoNLL 2004)*, pages 1-8.

Tan, H., Zhao, T., Zheng, J. (2008). Identification of Chinese event and their argument roles. In *Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops (CITWORKSHOPS 2008)*, pages 14-19.